WHAT IS CLAIMED IS

1. A system for reliably receiving data, comprising:

a memory;

write logic configured to receive data and an unreliable clock signal and write the data to the memory using the unreliable clock signal; and

read logic configured to generate a gapped clock signal and read the data from the memory using the gapped clock signal, the gapped clock signal being generated by turning on and off a constant local clock signal.

2. The system of claim 1, wherein the write logic includes:

a register configured to buffer the data, and

a write pointer configured to generate an address for writing the data from the register into the memory.

3. The system of claim 1, wherein the memory includes a first-in, first-out memory.

4. The system claim 1, wherein the read logic includes:

a gapped clock generator configured to generate the gapped clock signal from the constant local clock signal.

5. The system claim 4, wherein the gapped clock generator includes:

a first state machine configured to generate an enable signal having at least two states, and

a second state machine configured to turn on and off the constant local clock signal based on the state of the enable signal to generate the gapped clock signal.

Sub A1

6. The system of claim 5, wherein the read logic further includes:

a component configured to determine whether the memory contains data.

7. The system of claim 6, wherein the component includes:

a comparator configured to determine whether the memory contains data by comparing a write address used by the write logic to access the memory to a read address used by the read logic to access the memory.

8. The system of claim 6, wherein the second state machine is configured to turn off the constant local clock signal when the memory contains no data.

9. The system of claim 1, wherein the unreliable clock signal operates at a frequency lower than a frequency of the constant local clock signal; and

wherein the read logic is configured to compensate for underflow conditions in the memory by turning off the constant local clock signal.

10. The system of claim 1, wherein the unreliable clock signal operates at a frequency higher than a frequency of the constant local clock signal; and

wherein the read logic is configured to generate an error signal when overflow conditions occur in the memory.

Sub #1

11. The system of claim 1, wherein the write logic receives no unreliable clock signal; and

wherein the read logic is configured to start a counter and turn off the constant local clock signal.

12. The system of claim 11, wherein the read logic is further configured to determine that the write logic has received the unreliable clock signal before the counter reaches a predetermined count and turn on the constant local clock signal.

13. The system of claim 11, wherein the read logic is further configured to determine that the counter has reached a predetermined count and turn on the constant local clock signal.

14. The system of claim 13, wherein the read logic is further configured to wait for the write logic to receive the unreliable clock signal before reading data from the memory.

15. A system for reliably receiving data, comprising:

means for receiving data and an unreliable clock signal;

means for writing the data to a memory using the unreliable clock signal;

means for generating a gapped clock signal to compensate for underflow conditions in the memory; and

means for reading the data from the memory using the gapped clock signal.

Sub A1

16. A method for recovering data, comprising:

receiving data and an unreliable clock signal;

writing the data to a memory using the unreliable clock signal;

generating a gapped clock signal by turning on and off a constant local clock signal; and

reading the data from the memory using the gapped clock signal.

17. The method of claim 16, wherein the writing includes:

generating an address for writing the data into the memory.

18. The method of claim 16, wherein the generating includes:

generating an enable signal having at least two states, and

turning on and off the constant local clock signal based on the state of the enable signal to generate the gapped clock signal.

19. The method of claim 18, further comprising:

determining whether the memory contains data.

20. The method of claim 19, wherein the determining includes:

comparing a write address used to access the memory to a read address used to access the memory to determine whether the memory contains data.

21. The method of claim 19, wherein the turning includes:

stopping the constant local clock signal when the memory contains no data.

22. The method of claim 16, wherein the unreliable clock signal operates at a frequency lower than a frequency of the constant local clock signal; and

wherein the generating includes:

compensating for underflow conditions in the memory by turning off the constant local clock signal.

23. The method of claim 16, wherein the unreliable clock signal operates at a frequency higher than a frequency of the constant local clock signal; and

wherein the generating includes:

generating an error signal when an overflow condition occurs in the memory.

24. The method of claim 16, wherein the receiving includes:

receiving no unreliable clock signal; and

wherein the generating includes:

starting a counter when no unreliable clock signal is received, and

turning off the constant local clock signal when no unreliable clock signal is received.

25. The method of claim 24, wherein the generating further includes:

determining that the unreliable clock signal has been received, and

turning on the constant local clock signal when the unreliable clock signal has been received before the counter reaches a predetermined count.

26. The method of claim 24, wherein the generating further includes:

determining that the counter has reached a predetermined count, and

turning on the constant local clock signal.

27. The method of claim 26, wherein the reading includes:

waiting for the unreliable clock signal to be received before reading data from the memory.

28. A receiver, comprising:

a receiver component; and

a reliable clock generator configured to receive data and an unreliable clock signal, write the data to a memory using the unreliable clock signal, generate a reliable clock signal to compensate for underflow conditions in the memory, read the data from the memory using the

reliable clock signal, and provide the data and the reliable clock signal to the receiver component.

29. The receiver of claim 28, wherein the reliable clock generator includes:

a first state machine configured to generate first and second enable signals,

a second state machine configured to generate the reliable clock signal in response to the first enable signal, and

a register configured to receive the data read from the memory in response to the second enable signal.

30. The receiver of claim 29, wherein the reliable clock generator further includes:

a component configured to determine whether the memory contains data.

31. The receiver of claim 30, wherein the first state machine is configured to generate the first and second enable signals when the memory contains data.

32. The receiver of claim 30, wherein first state machine is configured to generate neither of the first and second enable signals when the memory contains no data.

33. The receiver of claim 32, wherein the first state machine is further configured to start a counter when the memory contains no data.

34. The receiver of claim 33, wherein the first state machine is further configured to generate the first and second enable signals when the memory contains data before the counter reaches a predetermined count.

Sub A1

35. The receiver of claim 33, wherein the first state machine is configured to determine whether the counter has reached a predetermined count and generate the first enable signal when the counter reaches the predetermined count.

36. The receiver of claim 29, wherein the second state machine is configured to toggle a constant local clock signal based on the first enable signal to generate the reliable clock signal.

37. A clock generator, comprising:

a first state machine configured to generate first and second enable signals, the first enable signal being used to read data from a memory that was written to the memory using an unreliable clock signal; and

a second state machine configured to generate a gapped clock signal for reliably recovering the data in response to the second enable signal.

38. The clock generator of claim 37, wherein the first state machine is further configured to determine whether the memory contains data.

39. The clock generator of claim 38, wherein the first state machine is configured to generate the first and second enable signals when the memory contains data.

Sub A1

40. The clock generator of claim 38, wherein the first state machine is configured to generate neither of the first and second enable signals when the memory contains no data.

41. The clock generator of claim 40, wherein the first state machine is further configured to start a counter when the memory contains no data.

42. The clock generator of claim 41, wherein the first state machine is further configured to generate the first and second enable signals when the memory contains data before the counter reaches a predetermined count.

43. The clock generator of claim 41, wherein the first state machine is configured to determine whether the counter has reached a predetermined count and generate the second enable signal when the counter reaches the predetermined count.

44. The clock generator of claim 37, wherein the second state machine is configured to toggle a constant local clock signal based on the second enable signal to generate the gapped clock signal.

Add A7